

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
4 September 2003 (04.09.2003)

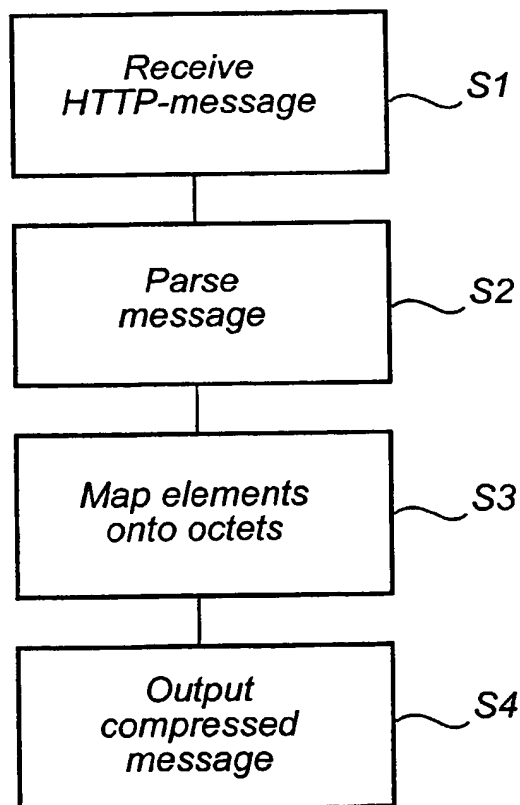
PCT

(10) International Publication Number
WO 03/073719 A1

- (51) International Patent Classification⁷: **H04L 29/00**
- (21) International Application Number: **PCT/IB02/00596**
- (22) International Filing Date: 28 February 2002 (28.02.2002)
- (25) Filing Language: English
- (26) Publication Language: English
- (71) Applicant (for all designated States except US): **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **RIIHJÄRVI, Janne** [FI/FI]; Valtatie 59 B 6, FIN-90500 Oulu (FI). **SHELBY, Zach** [FI/FI]; Loistotie 7, FIN-90520 Oulu (FI).
- (74) Agent: **AWAPATENT AB**; Box 11394, S-404 28 Göteborg (SE).
- (81) Designated States (national): AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, ES, FI (utility model), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK (utility model), SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: HTTP MESSAGE COMPRESSION



(57) Abstract: Method for compressing a HTTP message, including at least one field name and at least one field value, comprising parsing said HTTP message, to identify said at least one field name and said at least one field value, mapping each field name onto at least one binary octet (byte), the most significant bit (MSB) of said octet being set to "one", mapping each field values onto at least one binary octet (byte), the most significant bit (MSB) of said octet being set to "zero", and outputting said binary octets (bytes) to provide the HTTP message in compressed format. The method uses binary tagging instead of complex compression algorithms, making it extremely efficient if the processing power requirements (low) or latency-time (low) is considered.

WO 03/073719 A1



Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR,
GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent
(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR,
NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *with international search report*

HTTP MESSAGE COMPRESSIONTechnical Field

The present invention relates to a method for compressing a HTTP-message.

5 Technical Background

The Hyper-Text Transfer Protocol (HTTP) is a text rich application protocol developed for moving documents across the World Wide Web. Small ubiquitous and pervasive computing devices and (wireless) sensors usually have
10 very limited processing power and only narrowband connectivity to a network. For this reason, compression of some kind is advocated.

The trend in the field has been to study only transmission protocol compression (e.g. IP header
15 compression). However, this is not enough, as HTTP (in the payload) will dominate the traffic overhead. Therefore, compression of HTTP, which is and will be used extensively for many ubiquitous and wireless applications, is required.

20 An example of a compression method which can be used for HTTP compression, is given in WO 00/67382. According to this method, the fields of a HTTP header are coded by means of code words. Although a HTTP message can be compressed with the described method, the compression is
25 insufficient, as the method is not specifically highly optimized for small devices and low bit-rate communication.

Summary Disclosure of the Invention

30 An object of the invention is to effectively compress the HTTP header, using very limited processing power and latency.

This and other objects are achieved with a method for compressing a http-message, including at least one

field name and at least one field value, comprising parsing said HTTP message, to identify said at least one field name and said at least one field value, mapping each field name onto at least one binary octet (byte), the most significant bit (MSB) of said octet being set to "one", mapping each field values onto at least one binary octet (byte), the most significant bit (MSB) of said octet being set to "zero", and outputting said binary octets (bytes) to provide the HTTP message in compressed format.

Thus, according to the invention, the MSB of each octet (byte) is used to indicate whether a particular octet relates to a field name or a field value. As the MSB indicates when the field-name ends, and respectively when the field-value ends, there is no need for separators such as ":" and CRLF. In addition, most field-values (such as language tags, character sets etc.) can be easily enumerated, with most common values fitting in the 0-127 range, so that the entire header field can often be compressed into just two octets. Even for free-formed field-values (such as strings occurring in the Host-header) no special encoding is required, as they often consist of alphanumeric characters which can be sent with seven bits using e.g. ASCII code.

The method uses binary tagging instead of complex compression algorithms, making it extremely efficient if the processing power requirements (low) or latency-time (low) is considered. Hence, the low processing power and latency requirements have been taken as priority compared with the traditional full text compression approach.

The most obvious advantage of the invention is the high level of compression achieved. Instead of using three octets for separators, usually at least one for white space, and 2-19 octets for field-name specification, only one octet is used. Even for field-values large compression factors are obtained for content

encoding, media types etc. Thus the overall compression factor is usually quite high.

Also parsing the compressed message is, in most cases, extremely simple compared to parsing the case-insensitive ASCII field-names. A parsing algorithm can
5 very easily distinguish between field names and field values, regardless of their length.

In order to get an apprehension of the improvements in compression rate, the method according to the
10 invention can be applied to the HTTP message illustrated on page 14-15 of WO 00/67382, hereby incorporated by reference. While the method according to WO 00/67382 results in a compression rate (percentage of original message length eliminated) of 64%, the method according
15 to the present invention results in a compression rate of 73%. Note, however, that these figures are only an example, and depend on the message to be compressed. Other examples can be found, where the improvement is significantly larger.

20 Currently, many devices on the Internet make use of proxies for various reasons. The smallest devices will especially be forced to use proxies, gateways, and/or split protocol stacks in the future. This is to add security, caching capability, or to provide addresses to
25 devices. The method according to the invention is easy to implement as part of this proxy approach. The proxy device will handle the most complex part of the algorithm. The compression can be implemented with simple look-up tables, with minimal complexity added to normal
30 parsing of the HTTP-message.

The invention offers an efficient way to enable the use of HTTP and all applications based thereon in very cost efficient devices, and the possibility to embed compression functionality into split protocol stack
35 communication paradigms. It is especially valuable for low communication speed links and small embedded devices/sensors.

As the method leads to more efficient packaging, and faster and less complex parsing, it is advantageously used in small devices.

The HTTP message can be a request message, including
5 a request method, a URI, and a http version identifier.
In this case, the method can comprise treating said
request method and said HTTP version identifier as a
field name, mapping them onto at least one binary octet
with its MSB being set to "one", and treating said URI as
10 a field value, mapping it onto at least one binary octet
with its MSB being set to "zero".

The URI can be mapped using conventional ASCII
characters, i.e. one octet (byte) for each character,
with the MSB set to "zero". However, it is also possible
15 to map particular parts of the URI, such as "HTTP://", or
entire URI:s, onto one single octet.

The HTTP message can also be a respond message,
including a http version identifier, a status code, and a
status message. The method can then comprise treating
20 said status code and said http version identifier as a
field name, mapping them onto at least one binary octet
with its MSB being set to "one", and treating said status
message as a field value, mapping it onto at least one
binary octet with its MSB being set to "zero".

25

Brief description of the drawings

A currently preferred embodiment of the present
invention will be described in the following with
reference to the appended figure, where fig 1 is a
30 schematic view of an environment where the method
according to the invention may be implemented and fig 2
is a flow chart of a method according to an embodiment of
the invention.

Detailed description of preferred embodiments

The following binary compression scheme is based on HTTP/1.1, however the same technique applies to older and future versions.

- 5 An HTTP-message consists of a start-line, message-header, and message-body. The disclosed invention is only concerned with compressing the start line and message header.

10 The message-header in HTTP/1.1 consists of fields of the form

field-name ":" [field-value]

with possibly some white space without semantic
15 content. Fields are separated by CRLF sequences.

According to the invention, each field-name is mapped to an octet with the most significant bit (MSB) set, while field values get mapped to sequences of octets with the highest bits set to zero. No CRLF is needed.

- 20 If, for example, the field name "Content-Length" is mapped to [10010011], the field

Content-Length: 8200 CRLF,

- 25 where CRLF indicates "cariage return", would be mapped to

[10010011] - Content length

[01000000] - 64

[00001000] - 8 (8200 = 64*128 + 8).

30

- With the MSB indicating a field name, seven bits remain for coding the field name itself, in other words the code will allow for 128 field names. In the case of full HTTP/1.1 there are only 47 predefined header field
35 names. If more that 128 distinct field-names need to be conveyed, multiple octets with MSB set could be concatenated.

A special octet, such as [11111111], can indicate the end of the message-header (this could be omitted if the message-body is empty), and some other special bit sequence, such as [10000000], could act as the ", " of
 5 http, if this is deemed necessary.

The start line of a HTTP message is different depending on whether the message is a request message or a respond message.

For requests, the start-line is of the form:
 10

Method SP Request-URI SP HTTP-Version CRLF,

where SP indicates "space" and CRLF indicates "carriage return".

15 The proposed compression scheme is to handle the method and the HTTP-Version (HTTP/1.1 in our case) as a combined field-name, and the Request-URI as the field value. Preferably, the first part of the field name octet (e.g. the six first bits) indicate the method, and the
 20 last part (e.g. the two last bits) indicate the HTTP version.

If GET is mapped onto [100001] and HTTP 1.1 is mapped onto [01], then, as an example,

25 *GET http://www.oulu.fi HTTP/1.1*

would become

	[10000101] - <i>GET and HTTP-version</i>
30	[01101000] - <i>h</i>
	[01110100] - <i>t</i>
	[01110100] - <i>t</i>
	[01110000] - <i>p</i>
	[00111010] - <i>:</i>
35	[00101111] - <i>/</i>
	[00101111] - <i>/</i>
	[01110111] - <i>w</i>

[01110111] - w

[01110111] - w

Alternatively, an optional shorthand can be adopted
5 for the most common protocol identifiers, such as
[11000001] for http://.

Further, it is possible for the proxy to define
shorthands for commonly used URIs of a device. Thus, if a
URI such as http://our.server/camera/current.html was
10 mapped onto [00000001], then

GET http://our.server/camera/current.html HTTP/1.1

could be compressed quite simply as

15

[10000101][00000001].

If more than 24 extension methods are needed, or a
new HTTP-version provides added functionality, the
20 combined method/version field-name could again span
multiple octets (with highest bits set to 1) to give
enough space for enumerating the new methods.

For responses, the start-line reads

25 *HTTP-Version SP Status-Code SP Status-message CRLF,*

where, again, SP indicates "space" and CRLF indicates
"carriage return"

The compression can again be achieved, for example,
30 by combining the HTTP-Version and Status-Code as a field-
name, and giving the Status-Message as an optional value
for that header.

With reference to fig 1, the method can
advantageously be implemented in the communication
35 between a client device 1 (such as a PDA or sensor) and a
proxy 2, located intermediately between the client 1 and
a network 3. The method may be implemented by software,

being run on microprocessors or -controllers in the proxy and device respectively, but it may equally well be implemented by programmable logic circuits (FPGA), electronic components, or as part of ASIC-circuitry.

5 With reference to fig 2, the proxy receives (S1) a HTTP message from the network, and parses it (S2) in order to identify the field names and field values. Note that, according to the preferred embodiment, the start line (request or response) is also identified as
10 comprising field name and field value, as was described above.

 In the next step (S3), the parsed elements are mapped onto binary octets (bytes) using e.g. look-up tables, and the compressed message is outputted (S4).

15 The client receives the compressed message, and can very effectively parse it and identify the HTTP elements using an identical set of look-up tables.

 A similar routine can be followed when sending HTTP messages from the client to the proxy. A HTTP message is
20 compressed by the client, and sent to the proxy. The compressed HTTP message will be received by the proxy, and decompressed using the same look-up tables.

 Alternatively, applications on the client side can be adapted to receive and generate HTTP messages directly
25 in compressed format, to save processing resources.

 The above description of a preferred embodiment is not intended to limit the scope of the appended claim, and many modifications will be apparent to the skilled person. For example, it is not necessary to use the MSB
30 as "recognition bit", indicating the occurrence of field names, but instead this can be coded in any other place.

CLAIMS

1. Method for compressing a HTTP message,
including at least one field name and at least one field
5 value, comprising
parsing said HTTP message, to identify said at least
one field name and said at least one field value,
mapping each field name onto at least one binary
octet (byte), the most significant bit (MSB) of said
10 octet being set to "one",
mapping each field values onto at least one binary
octet (byte), the most significant bit (MSB) of said
octet being set to "zero",
and outputting said binary octets (bytes) to provide
15 the HTTP message in compressed format.

2. Method according to claim 1, further comprising
mapping each field name into two octets, each having
their respective MSB set to "one".
20

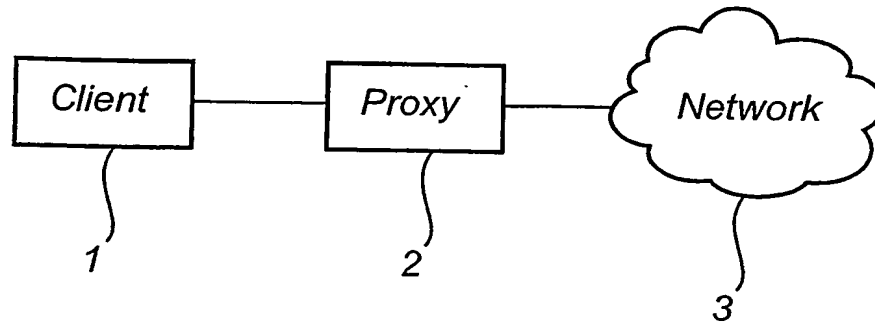
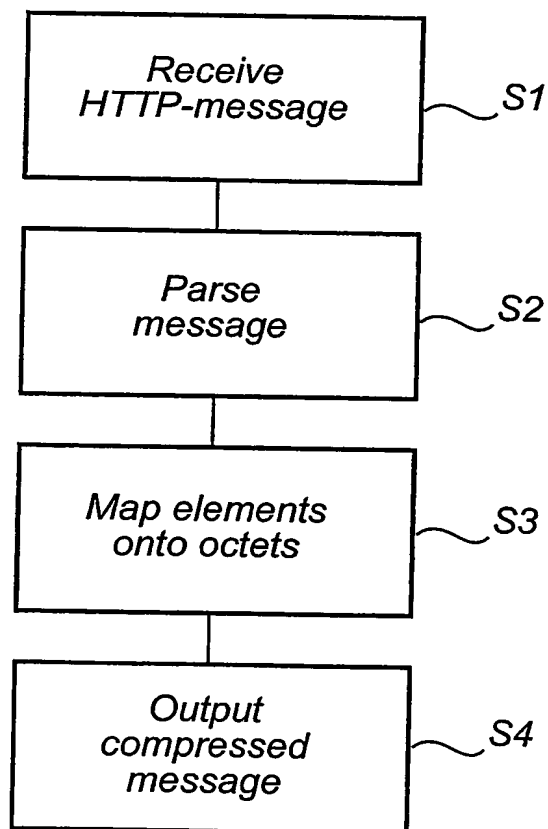
3. Method according to claim 1, wherein said HTTP
message is a request message, including a request method
identifier, a URI, and a HTTP version identifier,
comprising
25 identifying said request method identifier and said
HTTP version identifier as a field name, mapping them
onto at least one binary octet with its MSB being set to
"one", and
identifying said URI as a field value, mapping it
30 onto at least one binary octet with its MSB being set to
"zero".

4. Method according to claim 1, wherein said HTTP
message is a respond message, including a HTTP version
35 identifier, a status code, and a status message,
comprising
identifying said status code and said HTTP version

10

identifier as a field name, mapping them onto at least one binary octet with its MSB being set to "one", and identifying said status message as a field value, mapping it onto at least one binary octet with its MSB
5 being set to "zero".

1/1

*Fig. 1**Fig. 2*

INTERNATIONAL SEARCH REPORT

PCT/IB 02/00596

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04L 29/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI DATA, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0928070 A2 (UNWIRED PLANET, INC), 7 July 1999 (07.07.99), claims 12,17, [0062]	1-4
	--	
A	US 6286053 B1 (VAN PEURSEM, J.E. ET AL.), 4 Sept 2001 (04.09.01), column 3, line 1 - line 19; column 5, line 50 - line 61	1-4
	--	
A	US 5956490 A (BUCHHOLZ, D.R. ET AL.), 21 Sept 1999 (21.09.99), column 3, line 3 - line 30	1-4
	--	
A	WO 0119052 A2 (GENERAL INSTRUMENT CORP), 15 March 2001 (15.03.01), page 19, line 1 - line 6	1-4
	--	

☒ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family


Date of the actual completion of the international search

2 October 2002

Date of mailing of the international search report

28.10.2002

Name and mailing address of the International Searching Authority

 European Patent Office, P.B. 5818 Patentlaan 2
NL-2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Kristoffer Ogebjer/LR
Telephone No.

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

PCT/IB 02/00596

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6345307 B1 (BOOTH, R.C.), 5 February 2002 (05.02.02), table 2 -- -----	1-4

Form PCT/ISA/210 (continuation of second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

Information on patent family members

02/09/02

PCT/IB 02/00596

Patent document cited in search report			Publication date	Patent family member(s)		Publication date
EP	0928070	A2	07/07/99	JP	11284517 A	15/10/99
US	6286053	B1	04/09/01	NONE		
US	5956490	A	21/09/99	NONE		
WO	0119052	A2	15/03/01	AU	8035100 A	10/04/01
US	6345307	B1	05/02/02	AU	4226800 A	17/11/00
				WO	0067382 A	09/11/00
				AU	4472600 A	17/11/00
				EP	1175764 A	30/01/02
				WO	0067444 A	09/11/00

Form PCT/ISA/210 (patent family annex) (July 1998)